

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE
APPLIQUÉES

PAR
CHARLES SIROIS

DÉTECTION DES MAUVAISES HERBES DANS LES CULTURES DU
BLEUETS NAINS

DÉCEMBRE 2020

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

Table des matières

Sommaire	iv
Chapitre 1 Introduction	1
1.1 Objectifs du mémoire	2
1.2 Conclusion	3
Chapitre 2 Revue de littérature	4
2.1 Culture du bleuet nain et gestion des mauvaises herbes	4
2.2 Techniques de détection des mauvaises herbes par l'analyse d'images	6
2.3 Bibliothèques utilisées	10
2.4 Conclusion	14
Chapitre 3 Méthodologie	15
3.1 Acquisitions d'images	15
3.2 Traitements d'images	18
3.3 Classifications	23
3.4 Cartographie	26
3.5 Conclusion	29
Chapitre 4 Résultats	30
4.1 Indice de végétations	30
4.2 Classifications	32
4.3 Classifications avancées avec K-moyenne	35
4.4 Cartographie	36
4.5 Conclusion	38
Chapitre 5 Conclusion	39
5.1 Autres approches possibles	40

Table des figures

1	Cycle d'aménagement de bleuetière	4
2	Le kalmia à feuilles étroites	5
3	La comptonie voyageuse	6
4	Exemple de filtre <i>Excess of green</i>	7
5	Comparaison des méthodes de détections de points sur une image prise par Hubble.	11
6	Schéma de la méthodologie	15
7	Paramètres en jeux lors de l'acquisition d'images.	16
8	Préparation du drone pour l'acquisition d'images	17
9	Processus d'orthorectification	19
10	Orthophotographie produite par <i>OpenDroneMap</i>	20
11	Extrait d'image aérienne prise par un drone	21
12	Visualisation de l'application d'un indice de végétation	22
13	Comparaison d'une segmentation manuelle (milieu) à celle d'une segmentation otsu (droite) à partir du <i>ndvi</i> (gauche)	23
14	Extrait de parcelle dans le spectre du visible	25
15	Classification des pixels avec 2, 3 et 4 classes avec K-moyenne	26
16	Assimilation des classes intermédiaires à partir d'une classification à 4 classes	26
17	Orthophotographie produite par <i>OpenDroneMap</i>	28
18	Orthophotographie segmentée par la technique Yen	28
19	Carte de chaleur de la mauvaise herbe	29
20	Images des parcelles sélectionnées. a) Parcelle avec une haute densité de Kalmia b) Parcelle avec une faible densité de Kalmia c) Parcelle avec une densité moyenne de Kalmia d) Parcelle avec une densité moyenne de Kalmia e) Parcelle avec une haute densité de Kalmia f) Parcelle avec une densité moyenne de Kalmia	30
21	Application des indices de végétations <i>exg</i> , <i>exr</i> , <i>exgr</i> et <i>ndvi</i> sur les parcelles de A à F	31
22	Images de références des parcelles de A à F	32
23	Performance de différentes techniques de segmentation à partir de l'indice <i>exg</i>	32
24	Performance de différentes techniques de segmentation à partir de l'indice <i>exr</i>	33
25	Performance de différentes techniques de segmentation à partir de l'indice <i>exgr</i>	33
26	Performance de différentes techniques de segmentation à partir de l'indice <i>ndvi</i>	33
27	Performance de différentes techniques de segmentation à partir de l'indice <i>exg</i> sur 20 échantillons.	34
28	Image avec le filtre <i>ndvi</i> et sa segmentation de référence	35
29	Performance de segmentation selon le nombre de grappes	35
30	Performance de la nouvelle méthode des K-moyennes à partir de l'indice <i>exg</i> sur 20 échantillons.	36
31	Progression de l'infestation de la mauvaise herbe sur une période annuelle. a) 28 octobre 2019 b) 22 mai 2020 c) 9 novembre 2020	36
32	Carte de chaleur de la mauvaise herbe sur des images satellitaires de <i>Mapbox</i>	37

Liste des codes sources

1	Exemple d'utilisation d' <i>ODM</i>	14
2	Implémentation d'ODM	19
3	Implémentation des indices	21
4	Implémentation de la segmentation	24
5	Application de l'algorithme des K-moyennes	24

Sommaire

En agriculture, la gestion des mauvaises herbes est vitale pour l'agriculteur s'il souhaite répondre à la demande du marché tant au niveau des rendements des cultures récoltées, de la qualité de ces produits récoltés ainsi que des coûts de production des produits. Il existe plusieurs techniques de contrôle des mauvaises herbes comme l'analyse de végétales avec des images satellitaires. Dans un but d'optimiser le désherbage des champs, cette recherche va tenter de cartographier les mauvaises herbes comme le kalmia à feuilles étroites d'un champ de bleuet nain. Pour construire cette carte d'occupation de terrain des mauvaises herbes, des images des surfaces au sol seront prises à l'aide de drone muni d'une caméra multispectrale. Différents indices de végétations comme le *NDVI*, le *ExG*, le *ExR* ou le *ExGR* seront utilisés afin de différencier d'abord le sol de la végétation et ensuite les mauvaises herbes du bleuet nain. Le défi principal de cette recherche sera de faire la distinction entre les plants de bleuets et les plants de mauvaises herbes comme le kalmia et la comptonie.

Chapitre 1 Introduction

L'agriculture est l'un des piliers de la société d'aujourd'hui. Son optimisation est essentielle afin de maintenir ces solides assises. Le besoin grandissant de nourrir une population en perpétuelle expansion ont poussé les agriculteurs à devoir recourir à de nouvelles technologies d'optimisation. Ces pratiques ne sont pas toujours viables à long terme, car celles-ci viennent avec des compromis comme permettent une plus grande rentabilité pour répondre à la demande des consommateurs, mais laisse une empreinte environnementale. Ils existent plusieurs facteurs qui peuvent affecter la productivité d'un champ. L'une d'elles est la gestion des mauvaises herbes. Celles-ci ne sont pas toujours évidentes à contrôler selon le type de culture. Il existe plusieurs techniques pour gérer les mauvaises herbes d'un champ comme le désherbage manuel et l'utilisation d'herbicides et de pesticides. La technique la plus respectueuse de l'environnement est le désherbage manuel, car son mode opératoire est plus ciblé et n'introduit rien de nouveau dans l'écosystème du champ. Par contre, les cultures conventionnelles vont se tourner vers des alternatives moins coûteuses et plus rentables comme l'utilisation d'herbicides. Ce qui est tout à fait normal pour une entreprise qui ne veut pas faire faillite en produisant des cultures à très grande échelle avec des coûts de production moindres. Aujourd'hui, plusieurs outils existent, incluant les systèmes d'information géographiques qui prêtent mains fortes aux agriculteurs afin d'être mieux informés sur leurs champs. Ces systèmes de collectes de données sont plus accessibles, mais l'accès à des systèmes d'automatisation construit sur ces données restent à être développées.

La population est de plus en plus sensibilisée aux enjeux environnementaux. Cette sensibilisation engendre une demande grandissante pour des produits biologiques. Cette certification ajoute plusieurs restrictions à l'agriculteur, dont plusieurs concernant l'usage de certains herbicides. Le sarclage est une des options pour le cultivateur s'il veut vendre ces bleuets avec cette mention "biologique". Le cultivateur a plusieurs options pour vendre ses produits avec la mention biologique, mais l'obtention de celle-ci comporte un coût que le producteur doit

faire payer aux consommateurs. Malgré l'effort additionnel, ces techniques ont leurs limites vu qu'elles ne permettent pas l'éradication complète des herbes nuisibles. L'ajout d'automatisation dans ce processus pourrait rendre le contrôle des mauvaises herbes moins fastidieux, voire plus rentable. Le développement d'un processus de production de produits biologiques automatique ou semi-automatique permettrait de rendre ces produits plus accessibles.

1.1 Objectifs du mémoire

Cette recherche va se concentrer sur les champs de bleuets nains. Le projet de ce mémoire est d'explorer une solution pour automatiser une des facettes du processus d'élimination des mauvaises herbes du bleuet. La portée du projet va de l'acquisition des images jusqu'à la cartographie des plantes nuisibles. L'inspection automatique des champs de bleuet nain va produire une carte géoréférencée qui permettra d'orienter le cultivateur vers les surfaces à traiter dans sa gestion de mauvaises herbes. Car connaître l'emplacement précis des éléments nuisibles dans les champs est un atout à la fois économique et écologique. Ces deux atouts font *une pierre deux coups* pour l'agriculteur. D'une part, il réduit l'effort en désherbage manuel, d'autre part, il peut réduire la consommation d'herbicide en l'appliquant strictement là où il y a de la mauvaise herbe. Par conséquent, il taxe moins le sol avec des produits chimiques et il réduit ainsi des dépenses.

La structure de ce mémoire est scindée en plusieurs chapitres. D'abord, au chapitre 2, une revue de littérature pour présenter les articles et les bibliothèques utilisées qui ont servi à la réalisation du projet. Ensuite, au chapitre 3, la méthodologie qui est séparée en trois parties. La première partie couvre les méthodes utilisées pour acquérir les images à l'aide de drones et de différentes caméras. La deuxième partie explique les transformations diverses qui sont effectuées sur les images. La troisième partie montre les techniques utilisées pour différencier la mauvaise herbe des bleuets. Finalement, les résultats de la recherche sont présentés au chapitre 4, suivie d'une conclusion au chapitre 5.

1.2 Conclusion

En somme, il est essentiel d'explorer de nouvelles solutions pour optimiser l'agriculture. Une optimisation n'implique pas nécessairement un plus grand rendement au détriment de l'écosystème. En effet, s'il est possible d'automatiser la gestion de la mauvaise herbe, il est tout à fait possible de minimiser les impacts environnementaux. C'est pourquoi une cartographie des plantes nuisibles est proposée comme solution afin de résoudre cette problématique. Finalement les grandes lignes de ce document sont brièvement décrites.

Dans le prochain chapitre, la littérature sur les sujets présentés sera ausculter afin d'y trouver des techniques et des solutions pour arriver à nos buts. Par conséquent, des articles scientifiques, des libraires informatiques et des logiciels de traitement d'image seront examinés.

Chapitre 2 Revue de littérature

2.1 Culture du bleuet nain et gestion des mauvaises herbes

La variété du bleuet nain est une espèce sauvage où sa culture repose principalement sur la gestion de sa population donc de son occupation spatiales des superficies cultivées. Gagnon (2009) a produit un document qui couvre l'agriculture du bleuet nain au Saguenay-Lac-Saint-Jean. L'article mentionne des solutions aux agriculteurs pour optimiser leurs récoltes selon leurs types de cultures. Il y a deux type de culture : conventionnelle et biologique. Ces deux type de culture suivent un cycle bisannuel. Ce cycle est le plus rentable pour l'agriculteur, mais le contrôle de la mauvaise herbe demeure un défi. Durant le cycle de croissance du bleuet, il y a deux principaux plants qui font concurrence au bleuet : le kalmia à feuille étroite et la comptonie voyageuse. À l'instar du bleuet, ces plants aiment les sols acides et ils lui font compétition en le cachant du soleil.

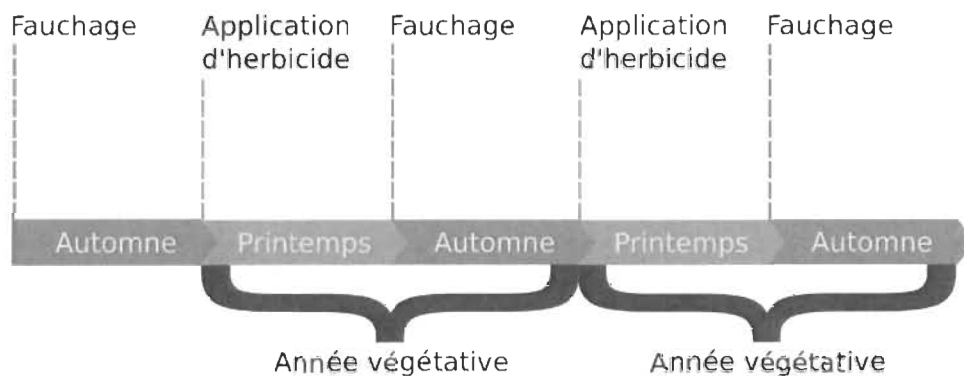


FIGURE 1 – Cycle d'aménagement de bleuetière

La nature sauvage de l'espèce du bleuet ne permet pas de faire une rotation des sols afin de contrôler ces plantes nuisibles. Il existe plusieurs produits pour contrôler ces mauvaises herbes, comme l'hexazinone et le glyphosate pour les cultures conventionnelles. Pour maximiser le contrôle des plants indésirables, ces herbicides sont utilisés conjointement avec un sarclage des champs lors de l'aménagement du champ comme illustré à la figure 1. Les opé-

ractions de sarclage et d'application d'herbicides sont des activités coûteuses pour les cultures conventionnelles et les cultures biologiques. Une solution automatisée pour le contrôle des mauvaises herbes allègerait ces coûts, peu importe le type de cultures.

2.1.1 Le kalmia à feuilles étroites



FIGURE 2 – Le kalmia à feuilles étroites

Connu sous plusieurs surnoms comme le crevard de mouton, *sheep laurel* ou *lambkill*, le kalmia à feuille étroite est un petit arbuste appartenant à la famille des *ericaceae*. L'origine de ces surnoms provient de la toxicité envers certains animaux dont le mouton. Décrit par Gagnon (2010b), cet arbuste peut atteindre 80 cm de hauteur et ces feuilles ont une forme elliptique d'une longueur de 2 à 9 cm. Cette plante fleurit de plusieurs petites fleurs roses durant le milieu de l'été.

2.1.2 La comptonie voyageuse

Expliqué par Gagnon (2010a), cette mauvaise herbe appartient à la famille des myricacées, la *comptonia peregrina* ou comptonie voyageuse est l'unique espèce de comptonie en Amérique du nord. Cet arbuste peut atteindre une hauteur d'environ 1 m et ces feuilles sont étroites et peuvent atteindre une longueur de 7 à 15 cm. Celle-ci pousse en agrégat grâce à ces rhizomes qui poussent horizontalement dans le sol.

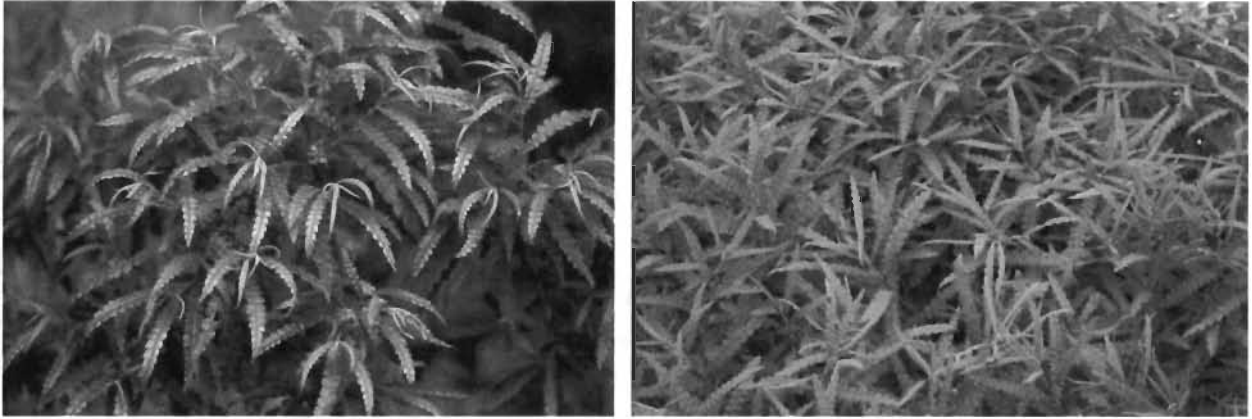


FIGURE 3 – La comptonie voyageuse

2.2 Techniques de détection des mauvaises herbes par l'analyse d'images

Ils existent plusieurs techniques pour détecter les plantes et ces méthodes appartiennent généralement à l'une de ces 3 approches de segmentation :

- à base de couleurs ;
- par seuillage ;
- et à base d'apprentissage.

Chacune de ces techniques suit une méthodologie similaire constitué de 2 parties : le pré-traitement et la classification de pixel. Le pré-traitement consiste à faire des ajustements sur les images, tels la réduction du bruit, l'ajustement du contraste et l'application de filtres. Ces ajustements initiaux ont tous des bénéfices selon le but de la classification.

2.2.1 Approche à base de couleur

L'approche à base de couleur consiste à faire des opérations sur les composantes de couleur. ming Li et al. (2009) détaille une transformation du modèle RVB (Rouge, Vert et Bleue) à TSL (Teinte, Saturation et Luminosité) où il n'y a que la composante teinte qui est utilisée pour procéder à la segmentation. Sinon, il existe d'autres styles d'opérations, comme les indices (ou index) de végétations. Certains indices se surpassent mieux que d'autres selon le type d'éclairage, le type d'image, le modèle colorimétrique et même le type de végétations. Hamuda et al. (2016) décrit une dizaine d'indices en expliquant brièvement le raisonnement

derrière le calcul. Chacune d'entre elles donne un résultat entre 0 et 1, où 0 indique le sol et 1 la végétation.



FIGURE 4 – Exemple de filtre *Excess of green*

Ces opérations peuvent être visualisées sur une image en nuance de gris comme à la figure 4. Les indices utilisés dans ce mémoire sont *Excess Green Index*, *Excess Red Index*, *Excess of Green Red* et le *Normalised Difference Vegetative Index*. L'indice de végétation *ExG* (*Excess Green Index*) et l'indice *ExR* (*Excess Red Index*) font tous les deux des manipulations dans le modèle colorimétrique RVB. *ExG* est un indice plutôt intuitif dans sa définition, en cherchant à amplifier la composante verte de l'image par deux. Alors que l'*ExR* s'inspire du fait que l'œil humain à une majorité de bâtonnet rouge soit 64%, donc on amplifie aussi cette composante par deux, mais selon le ratio de bâtonnet rouge. Ces deux indices se calculent respectivement par les formules :

$$ExG = 2v - r - b \quad \text{et} \quad ExR = 1.3r - v$$

où v , r et b sont respectivement la composante verte, rouge et bleue d'un pixel de l'image. L'indice *ExGR* (*Excess of Green Red*) combine les indices *ExG* et *ExR* via $ExGR = ExG - Exr$.

L'indice *NDVI* (*Normalised Difference Vegetative Index*), ou Indice de végétation par différence normalisée, opère sur les composantes du rouge et du proche infrarouge. Cet indice

exploite une des conséquences du phénomène de photosynthèse. En effet, les plantes émettent un surplus d'énergie solaire en radiation et celle-ci est détectable dans le proche infrarouge. Sa définition est

$$NDVI = \frac{pir - r}{pir + r}$$

où pir et r sont respectivement la composante proche infrarouge et rouge d'un pixel de l'image.

2.2.2 Approche par seuillage

L'approche par seuillage est un complément à l'approche à base de couleur. La plupart des indices donnent des résultats avec assez de nuance entre le sol et la végétation où une segmentation avec Otsu est suffisante. Cependant, il existe des techniques de segmentations plus élaborées permettant d'avoir une séparation du sol et de la végétation plus précise.

L'une de ces techniques est le seuillage automatique à plusieurs niveaux décrit par Yen et al. (1995). Cette technique consiste à trouver automatiquement le nombre de classification en minimisant une fonction de coût. Les critères de la fonction de coût sont construits sur deux facteurs. La première est la dissimilarité entre l'image originale et l'image segmenté, la seconde est le nombre de bit représentant l'image segmentée. Ces critères sont exprimés selon l'équation :

$$C(k) = \rho(Dis(k))^{1/2} + (\log_2(k))^2$$

où ρ est une constante positive, k est le nombre de classification et $Dis(k)$ est la dissimilarité. Quand le nombre de classification augmente, il réduit la dissimilitude et il augmente la nombre de bit représentant l'image segmentée. Donc cette équation, on recherche un compromis entre les deux facteurs qui la compose.

2.2.3 Approche à base d'apprentissage

Finalement, il y a l'approche à base d'apprentissage. Bien que les deux approches mentionnées plus tôt fonctionnent bien pour séparer le sol de la végétation, l'approche à base d'apprentissage vient combler certaines lacunes. L'une d'elles est le problème de la météo. En effet, la température n'est pas toujours clémente et celle-ci affecte la performance du seuillage. Une autre lacune est que les deux premières approches ne sont pas suffisantes pour dissocier les récoltes des mauvaises herbes.

Dans cette optique, il existe des techniques d'apprentissages pour aider à la segmentation ou à la classification. Certaines méthodes sont mentionnées par Hamuda et al. (2016) pour améliorer la segmentation des plantes du sol en temps nuageux ou ensoleillé. Par contre, ces techniques sont seulement utiles lorsque la prise d'images est faite sur une base régulière, donc à la merci de la météo. Pour la réalisation de ce projet, on a eu le luxe de choisir la journée pour faire le suivi des champs de bleuets. Puis il y a les méthodes de classification pour différencier les récoltes des mauvaises herbes. Par exemple, Sa et al. (2016) et dos Santos Ferreira et al. (2017) utilisent des réseaux neuronaux convolutifs pour identifier les plantes. La proximité des images prises dans ces deux articles sont relativement proches des plantes. C'est pourquoi l'utilisation de réseaux neuronaux est justifiée dans ces articles, car ceux-ci performant bien lorsqu'il y a beaucoup de nuances à extraire dans les caractéristiques de l'image.

Pour des images aériennes prises à plus haute altitude, l'information la plus intéressante à utiliser est l'indice de végétation. Bien que celle-ci sert principalement à séparer la végétation du sol, elle pourrait être utilisée pour identifier une plante en particulier. Par exemple, il existe des systèmes qui utilisent les signatures acoustiques des navires pour identifier leurs types de plates-formes. Dans un même ordre d'idée, on peut utiliser les indices de végétations comme une signature pour classer les plantes. Pour ces raisons, la technique des K-moyennes sera l'approche utilisée pour discerner les mauvaises herbes des bleuets.

2.3 Bibliothèques utilisées

2.3.1 *scikit-image*

van der Walt et al. (2014) ont initiés une bibliothèque d’algorithmes pour le traitement d’images en langage python intitulé *scikit-image*. Ce projet présente beaucoup de modules pour faire plusieurs opérations comme :

- la manipulation sur les bandes de couleurs ;
- l’application de filtres de segmentation ;
- des opérations de déformations ;
- et des algorithmes de détections d’objet.

scikit-image est un projet libre et ouvert dont le code est écrit et révisé par une communauté de volontaire. Les modules `filters`, `feature` et `metrics` sont les principaux modules utilisés dans ce projet.

Le module `filters` regroupe plusieurs implémentations de segmentation et de filtre. Deux fonctions sont utilisées conjointement ensemble : `threshold_yen` et `median`. La fonction `threshold_yen` applique un seuillage selon la méthode Yen et la fonction `median` retire le bruit d’une image à l’aide d’un filtre. La fonction `median` peut prendre en paramètre une matrice de 0 et de 1 ayant la morphologie du filtre utilisé par la fonction.

$$\text{disk}(2) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Cette matrice peut se générer aisément en utilisant un autre module de *scikit-image*. Le module `morphology` offre plusieurs fonctions utilitaires dont la fonction `disk`. Celle-ci génère une matrice de 0 et de 1 ayant la forme d’un disque avec un rayon paramétrable.

Le second module `feature` réunit plusieurs fonctions de reconnaissances de forme. Trois fonctions sont utilisées pour ce projet : `blob_dog`, `blob_doh` et `blob_log`. Ces fonctions permettent de faire la détection de points brillants sur des fonds sombres. Chacune d'elles utilise un algorithme différent avec des avantages et des désavantages différents.

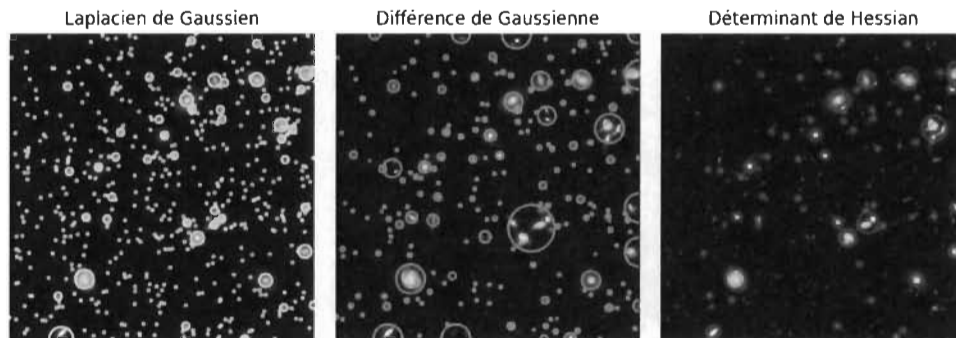


FIGURE 5 – Comparaison des méthodes de détections de points sur une image prise par Hubble.

La méthode `blob_log` utilise opérateur laplacien de la fonction gaussienne. C'est la méthode la plus précise, mais aussi la plus lente. La méthode `blob_dog` emploie une différence de gaussienne. Celle-ci est plus rapide que `blob_log`, mais ces deux méthodes souffrent d'un désavantage commun pour la détection de gros point. La méthode `blob_doh` se sert du déterminant de Hessian. C'est la méthode la plus rapide, car la vitesse de détection est indépendante de la grosseur des points sur l'image. Par contre, elle a des lacunes dans la détection de petit points. La figure 5 est un extrait de la documentation officielle de *scikit-learn*.

Le dernier module `metrics` regroupe un ensemble de méthodes pour calculer plusieurs métriques. La méthode utilisée dans ce mémoire est `adapted_rand_error`. Cette fonction prend en paramètre deux tableaux de nombre entier. Les tableaux de nombre entier dans ce cas-ci sont des images. La première image est une référence qu'on considère la plus proche de la vérité. La seconde image est celle qu'on veut comparer à la première pour mesurer sa similarité.

L'erreur adaptée de Rand retourne trois valeurs : la précision, le rappel et l'erreur. La précision est le nombre de paire de pixel dans l'image de référence et l'image de test correctement identifié divisé par le nombre de pixel dans l'image de test. Le rappel est similaire à la précision, mais au lieu de diviser par le nombre pixel dans l'image de test, on prend ceux dans l'image de référence. Finalement, l'erreur s'évalue avec l'équation suivante :

$$erreur = 1 - \frac{2pr}{p + r}$$

où p est la précision et r est le rappel.

2.3.2 *scikit-learn*

scikit-learn est une librairie d'apprentissage automatique en python écrit par Pedregosa et al. (2011). Cette librairie offre plusieurs modules dont le module `cluster`. Ce module regroupe plusieurs algorithmes de partitionnement de données non supervisés. L'algorithme des K-moyennes fait partie des implémentations disponibles dans le module `cluster`.

Le fonctionnement de cette algorithme se divise en trois étapes. Premièrement, on choisit les centroïdes initiaux en choisissant k échantillons du jeu de données. Ensuite on itère sur les étapes 2 et 3. L'étape 2 assigne chaque échantillon au centroïde le plus près. L'étape 3 crée un nouveau centroïde en évaluant la moyenne de tous les échantillons assignés à l'ancien centroïde. Ensuite on évalue la différence entre l'ancien et le nouveau centroïde. Si cette différence atteint un certain seuil l'algorithme prend fin et tous les échantillons sont regroupés en k grappes.

2.3.3 Planificateurs de vol et orthorectifieur

Pour arriver à nos buts, il faut pouvoir automatiser les vols de drone. Il existe plusieurs logiciels pour planifier ces types de vol. Lors de la recherche, trois approches ont été étudiées.

La première approche a été l'utilisation de *Tower*, une application mobile développée par DroidPlanner. La planification des missions dans l'application se fait entièrement manuelle-

ment au niveau de l'interface. Il est possible d'exporter et d'importer les missions sous des formats plus digérables pour l'automatisation. En prenant les positions géographiques des quatres coins d'un champ de bleuets, on peut générer l'itinéraire du drone. Cependant, le logiciel a eu sa dernière mise à jour en 2017 et semble avoir été abandonné depuis.

La deuxième approche a été d'explorer l'interface de programmation d'application dronekit. Cette IPA permet de développer des logiciels pour commander des drones en python ou en android. Bien que cela nous permet de faire une application spécifique à la problématique, le développement complet d'une application sort de la portée du projet.

La troisième approche été l'utilisation de *Mission Planner* de Michael Osborne. Il permet de générer des itinéraires suivant des patrons de vol afin de couvrir les régions désirées. Le logiciel est stable et il est maintenu fréquemment par la communauté *ArduPilot*.

Ensuite il faut trouver une solution pour orthorectifier ces images. Plusieurs logiciels sont décrits et comparés par Gross (2015). Par contre, aucun des logiciels proposés dans ce document n'a de licence assez permissive pour construire un outil accessible. OpenDroneMap est un projet qui propose plusieurs modules différents pour produire des orthoimages. La principale différence entre les modules est la façon dont on interagit avec le système. Par exemple :

- *WebODM* offre une interface graphique permettant de visualiser les résultats directement dans un navigateur web ;
- *ODM* est une interface en ligne de commande ;
- *PyODM* est un kit de développement en python.

Ces modules donnent de la flexibilité pour expérimenter et implémenter des solutions. *WebODM* permet d'expérimenter rapidement différent paramètre d'orthophotographie. Alors qu'*ODM* et *PyODM* offre des solutions plus permanentes et plus personnalisées. De plus, la nature libre et ouverte de la licence *OpenDroneMap* fait de ce logiciel un choix plus intéressant pour des fins de recherche.

Le fonctionnement classique d'*OpenDroneMap* est qu'il prend une collection d'images de

format jpeg, tiff ou des points de contrôle au sol et tisse les images ensemble afin d'obtenir une orthophoto. Plusieurs paramètres peuvent être ajoutés comme la résolution du produit finale ou des transformations sur les images en entrée. *ODM* peut être utilisé avec le conteneur logiciel *docker* pour simplifier son installation et son utilisation.

```
docker run -ti --rm
  -v /home/youruser/datasets:/datasets
  -v /home/youruser/results:/results
  opendronemap/odm --project-path /home/youruser/datasets
```

Code source 1: Exemple d'utilisation d'*ODM*

Un exemple de commande avec *ODM* qui s'exécute avec *docker* est illustré au code source 1. Cette commande *shell* attache les répertoires des images et des résultats au conteneur logicielle d'*OpenDroneMap*.

2.4 Conclusion

Dans ce chapitre, on a vu le cycle de culture du bleuet nain et une description des plantes nuisibles de ce fruit. Ensuite, 3 types d'approches pour analyser les images dans la détection de mauvaises herbes ont été étalées. Ces approches exploitent différentes techniques dont les indices de végétations, le seuillage adaptatif et l'apprentissage automatique. D'autre part, les bibliothèques informatiques qui ont contribué à ce projet ont été énoncées. Celles-ci permettent d'utiliser plusieurs algorithmes et techniques appartenant aux domaines de la vision par ordinateur et d'apprentissages automatique. Finalement il y a eu la recherche de logiciels pour planifier les vols de drone et pour tisser ces images ensemble.

Dans le prochain chapitre, les concepts et les techniques qui ont été explorés plus tôt vont être mis en pratique. La méthodologie va présenter concrètement les étapes qui ont été prises pour arriver aux objectifs fixés au début du mémoire.

Chapitre 3 Méthodologie

Le flux de travail peut se décrire en trois segments distincts consécutifs : l'acquisition, le traitement et le classement des images. En premier lieu, il faut planifier les différents paramètres du drone afin d'acquérir les images aériennes du champ. En deuxième lieu, il faut traiter les images en appliquant un indice de végétation approprié et coudre les images ensemble pour obtenir une carte. Finalement, on va appliquer un seuillage sur les images pour faire ressortir la position de toutes les mauvaises herbes.

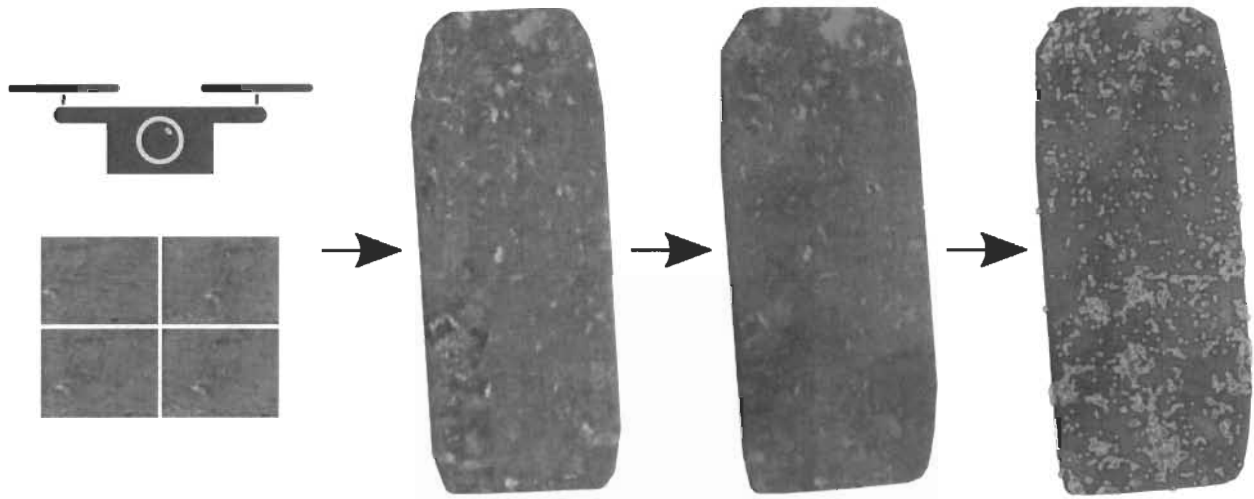


FIGURE 6 – Schéma de la méthodologie

Selon le type d'image, il va être possible d'ajouter une étape de plus en utilisant le module `feature` de la librairie *scikit-learn*. Celle-ci va nous permettre d'ajouter une position géographique à chacun des points trouvés.

3.1 Acquisitions d'images

La collecte d'image est réalisée à l'aide d'un drone muni d'une caméra. Cette caméra va prendre des images dans le spectre visible et le proche infrarouge. Les paramètres de vol

du drone seront importants à contrôler afin d'optimiser la qualité des images. Les multiples paramètres qui sont impliqués sont illustrés à la figure 7.

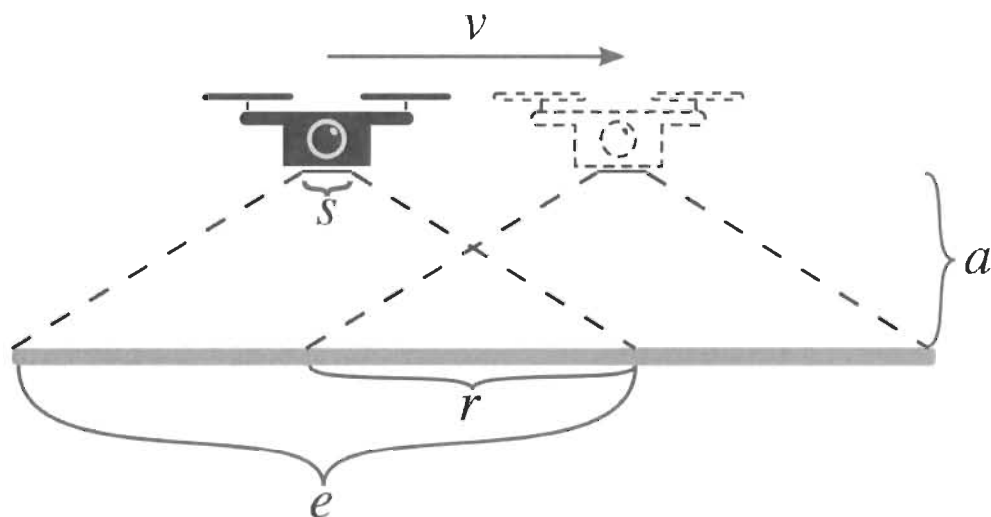


FIGURE 7 – Paramètres en jeu lors de l'acquisition d'images.

La vitesse du drone v doit être synchronisé avec sa caméra afin qu'il prenne des images qui ont un ratio de superposition r entre elles. L'importance de la superposition va être expliquée à la section 3.2. Pour évaluer la vitesse du drone, il faut d'abord savoir qu'elle est l'étendue e des images selon l'altitude a choisie. Le choix de la vitesse du drone doit être un juste milieu entre la clarté des images et l'économie de la batterie du drone. En effet, si le drone est trop rapide, les images risquent d'être floue ou trop sombre. Alors que si le drone est trop lent, il ne pourra pas couvrir beaucoup de surface et il dépensera plus de batterie.



FIGURE 8 – Préparation du drone pour l'acquisition d'images

Comme pour la vitesse, l'altitude du drone doit aussi être un juste milieu pour la visibilité des plantes. Si le drone vole trop près des plantes, le vent généré par les hélices du drone fait bouger les plantes et altère la clarté des images. La relation entre l'altitude a et l'étendue e peut être trouvée avec la distance focale f et les dimensions du capteur de la caméra s .

$$e_x = \frac{a}{f} s_x \quad , \quad e_y = \frac{a}{f} s_y$$

Selon la disposition de la caméra sur le drone, e_x ou e_y va être utilisé pour déterminer la vitesse du drone. Il existe un calcul un peu différent qui prend en compte l'angle du cardan qui tient la caméra. Cependant, la caméra a besoin d'avoir un angle perpendiculaire au sol si l'on veut produire une carte aérienne du champ. Cet angle perpendiculaire nous permet d'omettre ce paramètre dans le calcul de l'étendue d'image. Donc, en sachant l'étendue des images, le ratio de superposition et le nombre d'images pris par seconde (i_s), on peut déterminer la vitesse du drone via la formule

$$v = r i_s e$$

On peut remplacer e par sa définition pour obtenir la vitesse directement, à savoir

$$v = r \cdot i_s \cdot \frac{a}{f} \cdot s$$

Pour illustrer cette équation avec des valeurs plus concrètes, on peut utiliser les caractéristiques de la caméra *X3* du drone *Inspire 1*. Cette caméra a une distance focale de 4mm et la dimension de son capteur est de 6.17mm par 4.55mm et on assume que le drone se déplace dans la direction de la hauteur du capteur. Si le drone vole à une altitude de 10m avec une caméra qui prend 2 images par seconde avec un recouvrement de 70% entre chaque image, la vitesse correspondante est :

$$v = r \cdot i_s \cdot \frac{a}{f} \cdot s = 0.7 * 2 \cdot \frac{10}{4} \cdot 4.55 = 15.92 \text{ m/s}$$

3.2 Traitements d'images

Les images fournies par le drone vont subir plusieurs transformations. Ces traitements sont divisés en deux parties. Premièrement, les images vont être cousues ensemble afin d'obtenir une orthophotographie du champ. Deuxièmement, on applique un indice de végétation pour avoir des images avec des nuances de gris. Ces étapes sont interchangeables, mais durant les expérimentations on a remarqué que le processus d'orthophotographie avec *ODM* crée une diminution de la qualité des images. Par contre, la méthode change un peu avec les images multispectrales où on opère avec des images déjà orthophotographiées pour calculer les indices de végétations. Donc, le calcul d'indice de végétation se fait entre plusieurs images représentant les bandes désirées selon l'indice.

3.2.1 Orthophotographie

Un recouvrement large aide au succès d'une bonne orthophotographie. Illustré à la figure 9, la redondance de l'information par un recouvrement large s'observe tout de suite. Cette répétition de l'information permet une préservation de l'intégrité des données. Ainsi, on s'assure que l'information acquise par le drone ne s'est pas perdue en chemin.

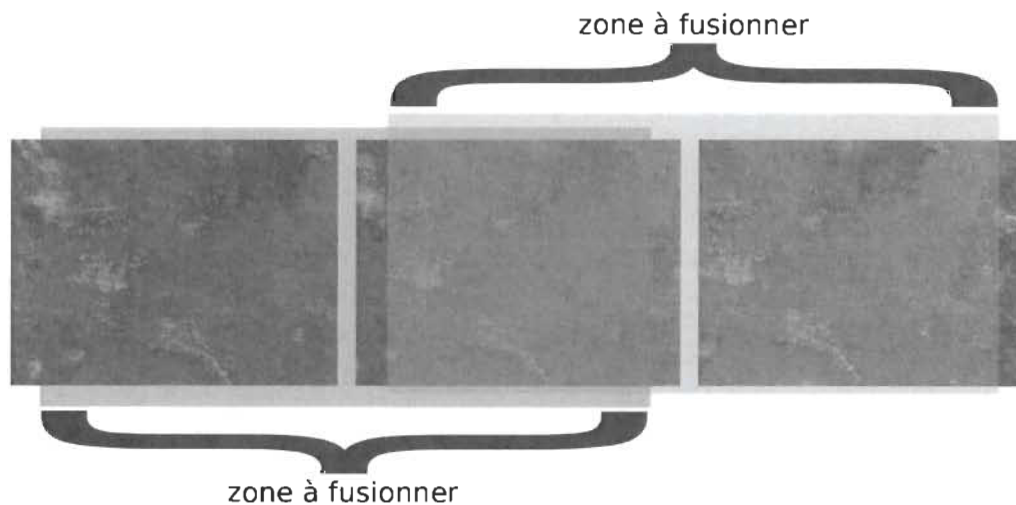


FIGURE 9 – Processus d'orthorectification

Par défaut, *OpenDroneMap* produit différents type de résultats, mais il n'y a que l'orthophoto qui nous intéresse pour ce projet. L'interface en ligne de commande permet l'ajout de plusieurs options dont le type de résultat recherché. Un script (code source 2) a été écrit pour englober la commande *odm* en n'exposant que les paramètres requis.

```
#!/bin/sh
docker run -it --rm
  -v "$(pwd)/$1:/code/code/images" \
  -v "$(pwd)/odm_orthophoto:/code/code/odm_orthophoto"
  opendronemap/odm --project-path .
  --orthophoto-resolution 2
```

Code source 2: Implémentation d'ODM

Selon la quantité d'images et la résolution sélectionnée, cette opération peut prendre plusieurs heures. Mais cette attente permet d'avoir une fondation solide pour la suite des transformations, tel qu'illustrée à la figure 10.

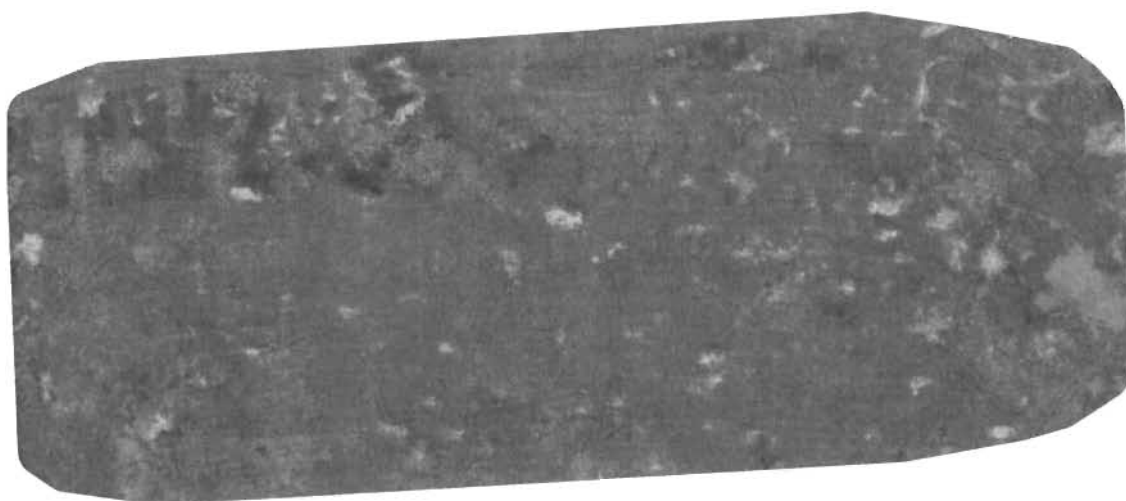


FIGURE 10 – Orthophotographie produite par *OpenDroneMap*

3.2.2 Indices de végétations

Les images peuvent être produites par différentes lentilles ou être acquises à différentes saisons de l'année. C'est pourquoi il est nécessaire de tester différents indices de végétations afin d'avoir les meilleurs résultats. Ainsi pour faciliter l'utilisation de différents indices de végétation, un script en python a été développé. Quatre indices de végétations sont implémentées dans le script : *ExG*, *ExR*, *ExR* et le *NDVI*.

L'implémentation du script va se traduire en une interface en ligne de commande. Cette décision nous permet d'expérimenter facilement les différents indices, mais aussi de permettre le traitement d'un grand volume d'images. L'interface va accepter un indice en option avec une liste de un ou plusieurs noms de fichier d'images. Pour chaque image traitée, il y aura une version de celle-ci sauvegardée avec l'indice de concaténer dans son nom de fichier. Le nom du programme est «vegutil», un mot-valise des mots anglais *vegetation* et *util*.

Le code source 3 est un extrait de *vegutil* qui définit une lambda pour chaque indice de végétation qui va opérer sur les composantes de couleurs. Une de ces lambdas définies est ensuite passé à la fonction `_convertier_template` pour appliquer l'indice de végétations sur tous les pixels. Cette structure modulaire du code permet l'ajouts de nouveaux indices de végétation sans trop de friction.

```

def exg(image):
    #  $2g - r - b$ 
    func = lambda colors: (2*colors[1] - colors[2] - colors[0])
    return _converter_template(image, func)

def exr(image):
    #  $1.3r - g$ 
    func = lambda colors: (1.3*colors[2] - colors[1])
    return _converter_template(image, func)

def exgr(image):
    #  $ExG - ExR$ 
    func = lambda cch: ((2*cch[1] - cch[2] - cch[0]) - (1.3*cch[2] - cch[1]))
    return _converter_template(image, func)

def ndvi(image):
    #  $\frac{NIR - Red}{NIR + Red}$ 
    func = lambda colors: ((colors[0] - colors[2]) / (colors[0] + colors[2]))
    return _converter_template(image, func)

```

Code source 3: Implémentation des indices



FIGURE 11 – Extrait d'image aérienne prise par un drone

Prenons par exemple une des images acquises lors de ce projet illustrée à la figure 11 pour démontrer le fonctionnement du programme. Avec la commande `vegutil -i exg exemple.jpg`, l'image est ouverte en un tableau *numpy* et on passe le tableau en paramètre à la méthode `apply_along_axis`. Cette méthode permet de parcourir un tableau selon un axe

choisi et d'y appliquer une fonction de transformation. Dans ce cas-ci, la fonction de transformation est déclarée selon les lambdas au code source 3 et l'axe choisi est la profondeur de l'image pour que la méthode `apply_along_axis` puisse fournir la valeur des couleurs aux lambdas. La figure 12 est une visualisation d'une des itérations de `apply_along_axis`.

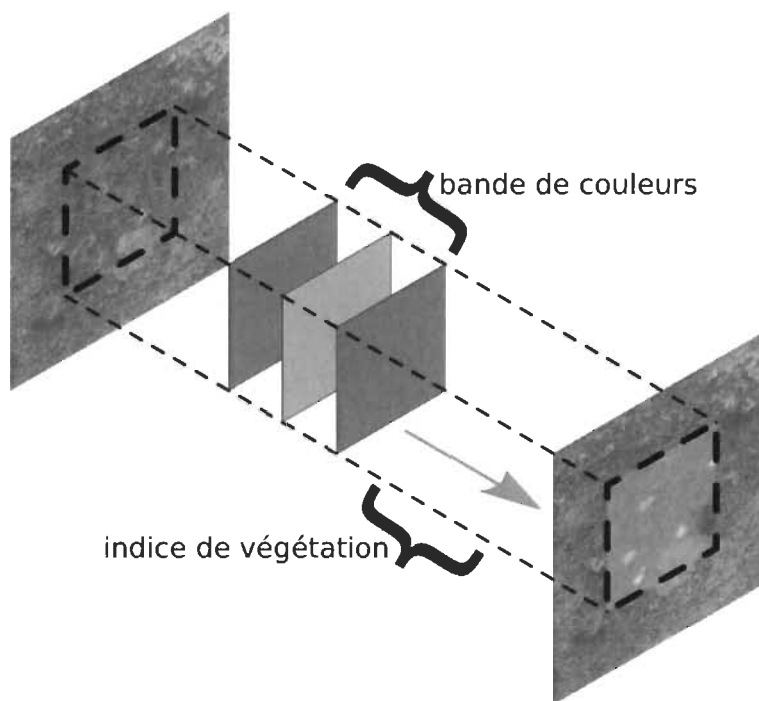


FIGURE 12 – Visualisation de l'application d'un indice de végétation

Donc on peut voir que le résultat de ce processus va nous retourner une image en nuance de gris où les pixels sombres représentent le sol et les pixels claires représentent la végétation. Pour les images multispectraux, la structure du code est similaire avec quelques différences comme les indices en entrée et la préparation des données. En fait, ces différences ont amenés à faire une version de *vegutil* qui ne traite que les images multispectraux. Nommées *vegutil-mi*, ce programme doit avoir en entrée chaque bande requise sous forme d'image pour évaluer l'indice. Un exemple d'utilisation ressemblerait à ceci : `vegutil-mi --indice exg --order red green blue --images red.tif green.tif blue.tif`. La suite est similaire à *vegutil* où les bandes de couleurs nécessaire sont fournies par plusieurs images au lieu d'une seul.

3.3 Classifications

3.3.1 Segmentations



FIGURE 13 – Comparaison d’une segmentation manuelle (milieu) à celle d’une segmentation otsu (droite) à partir du *ndvi* (gauche)

En explorant les signaux de végétation, il est possible de trouver une valeur qui semble correspondre à la mauvaise herbe. À la figure 13, l’image du milieu a été segmentée en choisissant une valeur de seuillage manuellement. À titre de comparaison, une segmentation avec la méthode de Otsu a été réalisée sur celle de droite. Empiriquement, on remarque que la méthode de Otsu est plus sensible dans sa segmentation.

Les opérations de segmentation ont été greffées à *vegutil* comme une option de plus en entrée. *vegutil* permet de spécifier deux style de segmentations : la méthode Yen et la méthode Otsu. En utilisant le module `filters` de *scikit image* et les opérateurs booléens sur les matrices de *numpy*, l’implémentation est relativement concise.

La méthode `threshold_yen` et la méthode `threshold_otsu` du code source 4 retourne la valeur supérieure du seuil. Les matrices *numpy* permettent de faire des comparaisons booléennes avec des valeurs scalaires. Celle-ci retourne une matrice booléenne représentant l’image segmentée. La fonction `filters.median(result, dis(3))` retire tous les éléments de l’image qui sont plus petits qu’un disque de 3 pixels de rayon. Ceci permet de nettoyer le

```
def apply_threshold(image, method):
    if method == 'yen':
        binary = image >= filters.threshold_yen(image)
    elif method == 'otsu':
        binary = image >= filters.threshold_otsu(image)

    result = img_as_uint(binary).astype('uint8')
    result = filters.median(result, disk(3))
    return result
```

Code source 4: Implémentation de la segmentation

bruit dans l'image.

3.3.2 K-moyennes

L'utilisation de l'algorithme des K-moyennes permet d'avoir un peu plus de nuances dans la classification de la végétation. Bien que les images sont prises aux moments où seule la mauvaise herbe est verte dans les champs, les plants de bleuets émettent quand même un signal dans les indices de végétations. Les images sont soumises à plusieurs transformations avant de créer le modèle de partitionnement de données. Il est aussi possible d'ajouter de la dimensionnalité aux données en combinant plusieurs couleurs ou indices.

```
import numpy as np
from skimage import io
from sklearn.cluster import KMeans

vert = io.imread('vert.tif')
pir = io.imread('pir.tif')

vert = vert / np.linalg.norm(vert)
pir = pir / np.linalg.norm(pir)

stacked = np.dstack((vert, pir))

X = stacked.reshape((-1, 2))

kmeans = KMeans(n_clusters=3)
kmeans.fit(X)
```

Code source 5: Application de l'algorithme des K-moyennes

Au code source 5, *numpy* est utilisé pour normaliser les valeurs de réflectance des images et pour les empiler (ajout de dimensionnalité aux données) avant de construire le modèle. Donc ici l'hypothèse est que sachant qu'il y a 3 éléments à classer sur l'image (sol, bleuets et mauvaise herbe), il devrait avoir assez d'information pour qu'un modèle de K-moyennes puisse bien les identifier. Cette technique a une implémentation différente dans les résultats à la section 4.2 où on met le nombre de classe à 2 avec pour seule dimension les images avec le filtre d'indice de végétation.

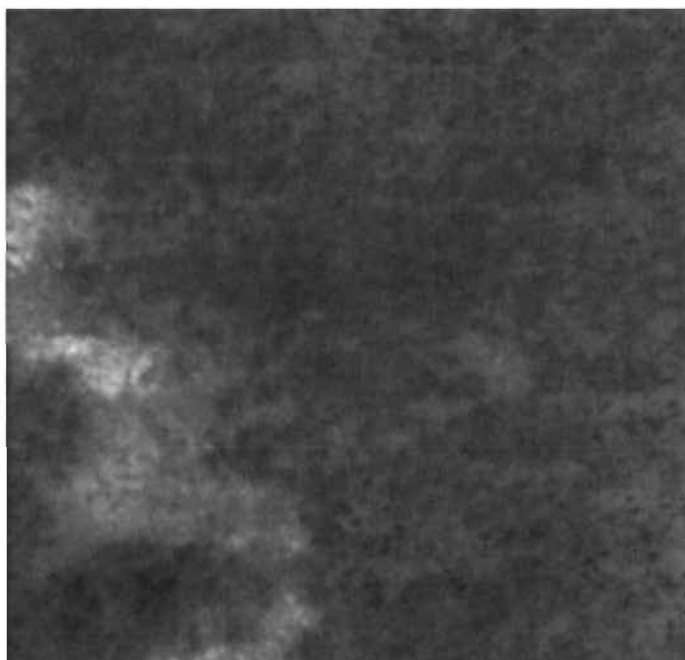


FIGURE 14 – Extrait de parcelle dans le spectre du visible

Par contre, à la section 4.3, on va explorer seulement la méthode des K-moyennes en variant le nombre de classes. Étant donné que les indices de végétation classent la valeur des pixels entre 0 et 1, il est possible d'utiliser la valeur des centroïdes du modèle des K-moyennes pour déterminer quelles classes appartiennent au sol ou à la végétation. C'est pourquoi on assume que les extrémités sont le sol et la mauvaise herbe et que les classes intermédiaires sont potentiellement les plants de bleuets. La figure 15 illustre le nuancement des classifications à partir de la figure 14 lorsqu'on augmente le nombre de grappe du modèle de K-moyenne.

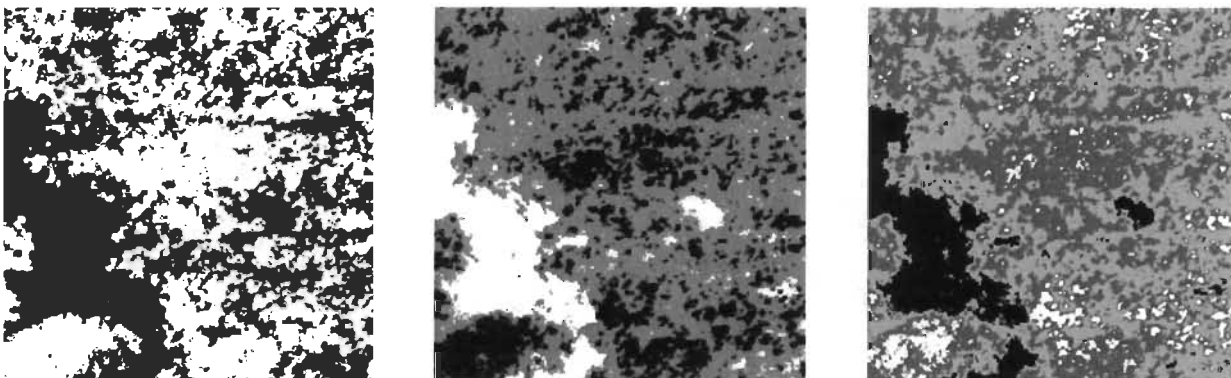


FIGURE 15 – Classification des pixels avec 2, 3 et 4 classes avec K-moyenne

Afin d'isoler la mauvaise herbe, il est possible d'étiqueter les classes intermédiaires comme étant l'arrière-plan de l'image. Cette opération fusionne les plants de bleuets avec le sol. Ainsi on se retrouve avec une image où seul la mauvaise herbe est segmentée. La figure 16 montre ce processus d'assimilation où on commence avec 4 classes pour ensuite assimiler les 2 classes intermédiaires pour finalement les étiquetées comme arrière-plan.

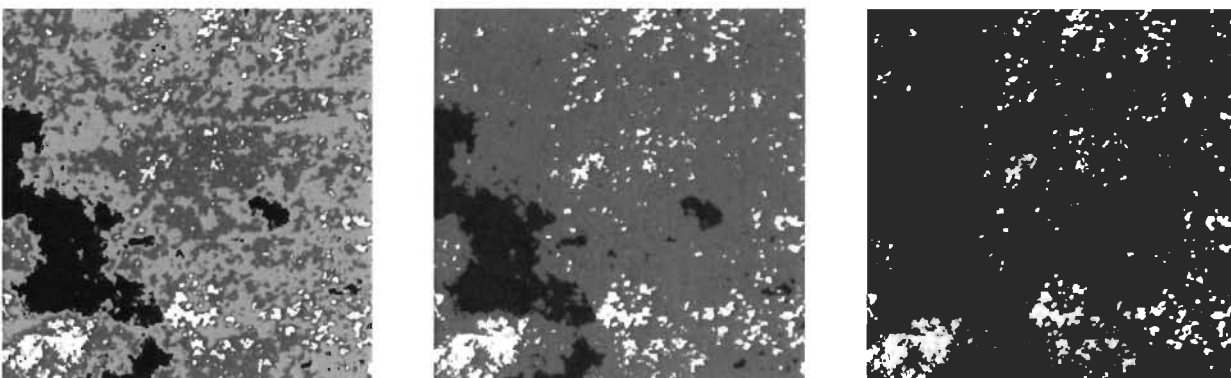


FIGURE 16 – Assimilation des classes intermédiaires à partir d'une classification à 4 classes

3.4 Cartographie

Avec les résultats de la classification, il est possible de les transformer en des données plus digestibles pour des systèmes informatiques ou pour des agriculteurs. Toujours dans le langage python, un programme a été écrit pour identifier la position des points correspondant

à la mauvaise herbe sur les images. Dans le même style de *vegutil*, ce programme est une interface en ligne de commande prenant plusieurs options en entrée. Ce programme prend une image segmentée dans un format où il contient des métadonnées géographiques comme le format d'image GeoTIFF avec une méthode de détection et une méthode de sortie. Les méthodes de détection utilisent le module **feature** de *scikit-image* expliqué à la section 2.3.1. Celle-ci permet de spécifier quels styles de détection à utiliser soit :

- *log*, laplacien de la fonction gaussienne ;
- *dog*, différence de gaussienne ;
- *doh*, déterminant de Hessian.

Les méthodes de sortie vont déterminer le format dans lequel il va être consulté. Présentement, le programme offre 4 méthodes de sortie :

- *circle*, chaque point est entouré d'un cercle rouge ;
- *heat*, une carte de chaleur représentant la densité de points sur l'image ;
- *map*, produit un fichier *index.html* affichant une carte de chaleur sur une carte géographique ;
- *raw*, produit un fichier contenant les données brutes de la position des points dans le format latitude longitude.

Par exemple, la figure 17 est une orthophotographie produite avec *OpenDroneMap* en utilisant des images acquises par le drone. Les tuiles utilisées pour l'orthophoto ont été pré-traitées avec l'indice de végétation *exg*.

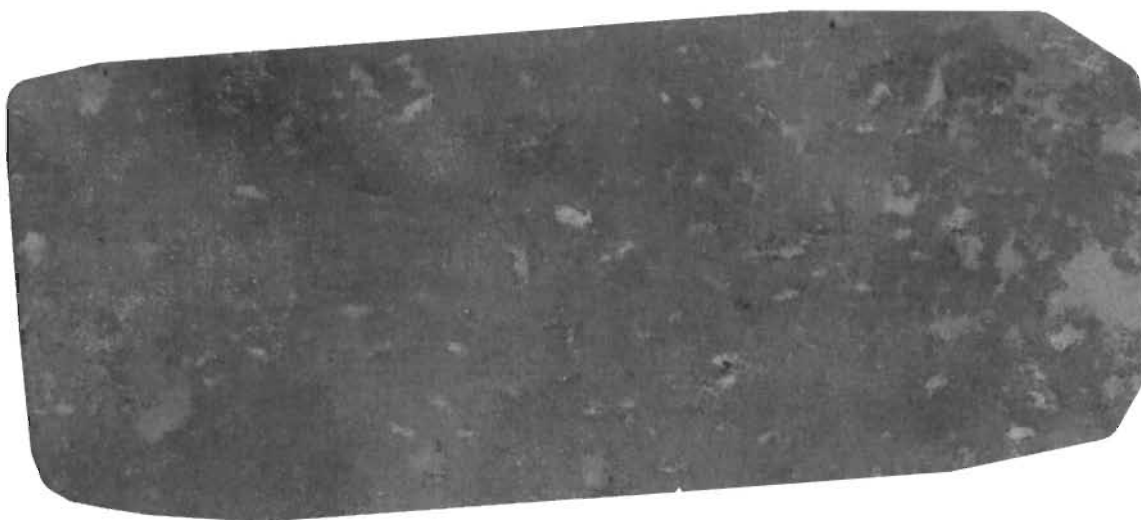


FIGURE 17 – Orthophotographie produite par *OpenDroneMap*

Toutefois cette image ne donne pas beaucoup d'information, mais en utilisant la technique de segmentation de Yen on peut commencer à voir où se trouve la mauvaise herbe à la figure 18. Ensuite avec l'utilisation du programme *blober*, on peut prendre l'information de la figure 18 et la représenter comme une carte de chaleur sur la figure 17. Grâce à ce procédé, on peut voir que le champ est particulièrement affecté du côté gauche de la figure 19.



FIGURE 18 – Orthophotographie segmentée par la technique Yen

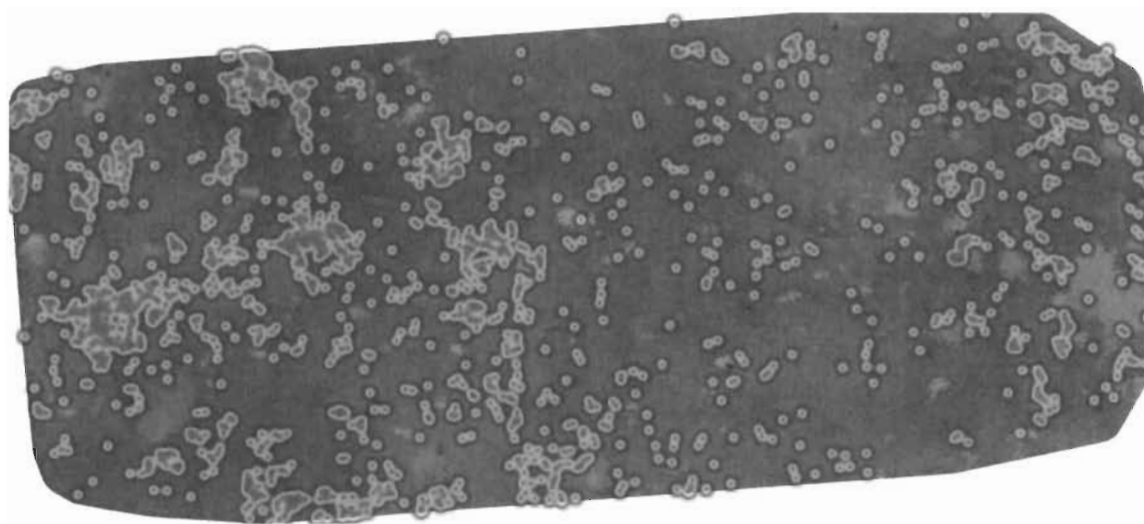


FIGURE 19 – Carte de chaleur de la mauvaise herbe

3.5 Conclusion

En résumé, la méthodologie est constituée de 4 parties, dont les 3 premières sont cruciales pour le succès de la dernière partie. D'abord il faut établir un processus d'acquisition d'images balançant l'optimisation de la batterie du drone et la qualité des images prises. Ensuite il y a le traitement d'image se scindant en 2 sous parties : l'orthorectification des images et l'application de l'indice de végétation. Ces étapes vont nous permettent d'avoir une carte du champ ayant comme information l'intensité végétative. Puis la classification est l'étape où on passe d'une information analogue à binaire à l'aide de technique de seuillage. C'est-à-dire l'étiquetage de deux valeurs possibles dans les orthophotographies, soit le sol ou la mauvaise herbe. Le succès de cette étape va nous permettre de dresser une vue d'ensemble précise de l'infestation d'un champ de bleuet nain.

Dans le chapitre 4, la méthodologie va être mise à l'épreuve. L'aboutissement des 3 premières parties de ce chapitre sera mesuré à l'aide du calcul de l'erreur de Rand adapté. Donc c'est dans ce chapitre qu'on va déterminer quelles techniques de classification performant mieux que les autres.

Chapitre 4 Résultats

Les résultats sont présentés dans l'ordre logique qu'ils ont été amenés au chapitre 3. La première section montre les transformations des images brutes vers les divers indices de végétations. La deuxième section présente plusieurs graphiques comparant les performances de seuillage selon l'indice de végétation. La troisième section est l'exploration des variations du nombre de grappes pour les modèles de K-moyennes. Finalement, la quatrième section présente des utilisations possibles des résultats obtenus dans les 2 premières sections.

4.1 Indice de végétations

Les analyses sont réalisées principalement sur des images multispectrales prises le 22 mai 2020 au nord de La Tuque. Afin de faciliter les mesures de performances, 6 parcelles prédéfinies sont employées pour les tests.

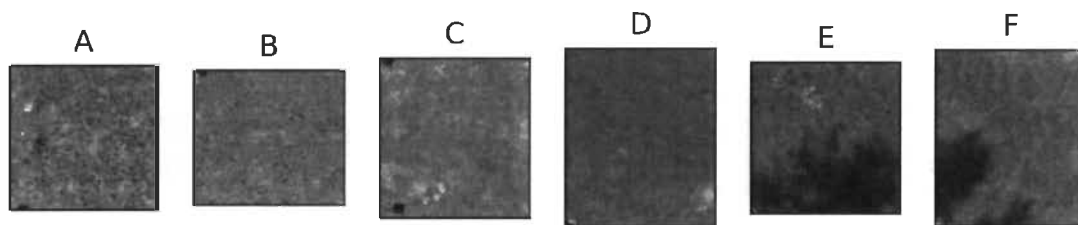


FIGURE 20 – Images des parcelles sélectionnées. a) Parcelle avec une haute densité de Kalmia b) Parcelle avec une faible densité de Kalmia c) Parcelle avec une densité moyenne de Kalmia d) Parcelle avec une densité moyenne de Kalmia e) Parcelle avec une haute densité de Kalmia f) Parcelle avec une densité moyenne de Kalmia

Dans un désir de visualisation de ces parcelles à la figure 20, la bande rouge, verte et bleue ont été superposées. La coloration n'est pas parfaite, mais la végétation est assez visible pour déterminer si nos méthodes fonctionnent bien.

Par contre, le calcul des indices de végétation est capable de nuancer ce genre d'information

comme on peut voir à la figure 21. La végétation dans les parcelles A, B et C se font relativement bien détecter par les indices de végétations. On remarque que l'*exr* fait l'inverse des autres indices en ayant la végétation sombre et le sol clair. Néanmoins les mesures de précision de seuillage à partir de l'*exr* n'est pas affecté par cette particularité. Aussi le *ndvi* semble plus performant pour faire ressortir les nuances des signaux dans les parcelles D, E, F.

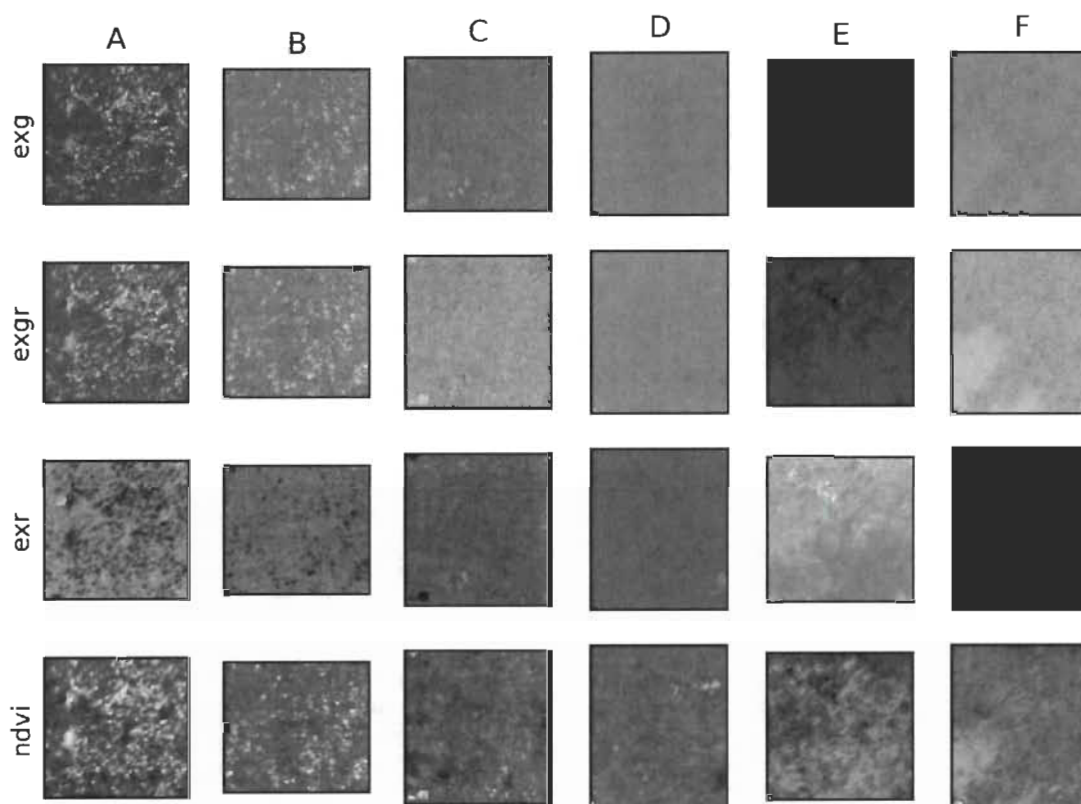


FIGURE 21 – Application des indices de végétations *exg*, *exr*, *exgr* et *ndvi* sur les parcelles de A à F

4.2 Classifications

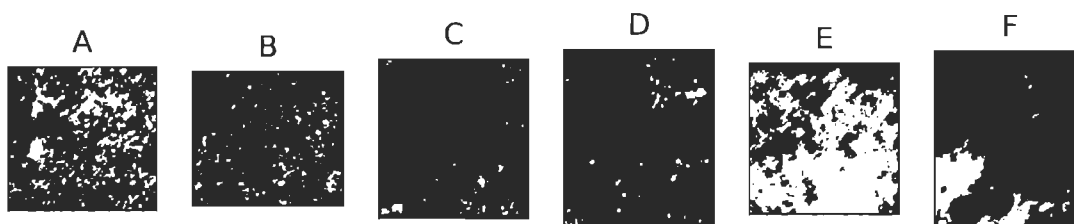


FIGURE 22 – Images de références des parcelles de A à F

Les images de références ont été produites manuellement dans le logiciel *Gimp* (*GNU Image Manipulation Program*) à partir des résultats du *ndvi*. Pour chaque parcelle de chaque indice de végétation, on segmente l'image en utilisant 3 techniques différentes. Ensuite, chacune des segmentations est comparée aux images de références avec la méthode `adapted_rand_error` de *scikit-image*. Celle-ci retourne 3 valeurs expliquées à la section 2.2.1. Une moyenne pour chacune de ces valeurs est évaluée à partir des 6 résultats et elle sont présentées aux figures 23, 24, 25 et 26.

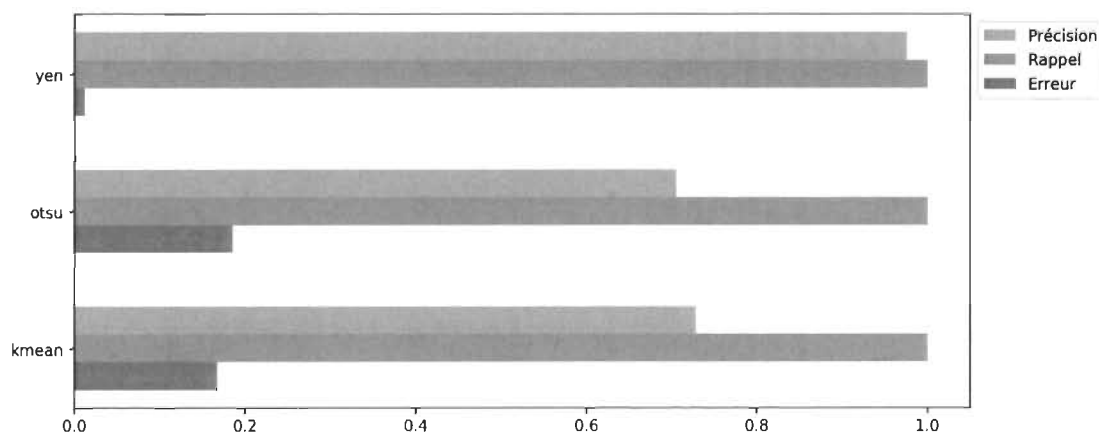


FIGURE 23 – Performance de différentes techniques de segmentation à partir de l'indice *exg*

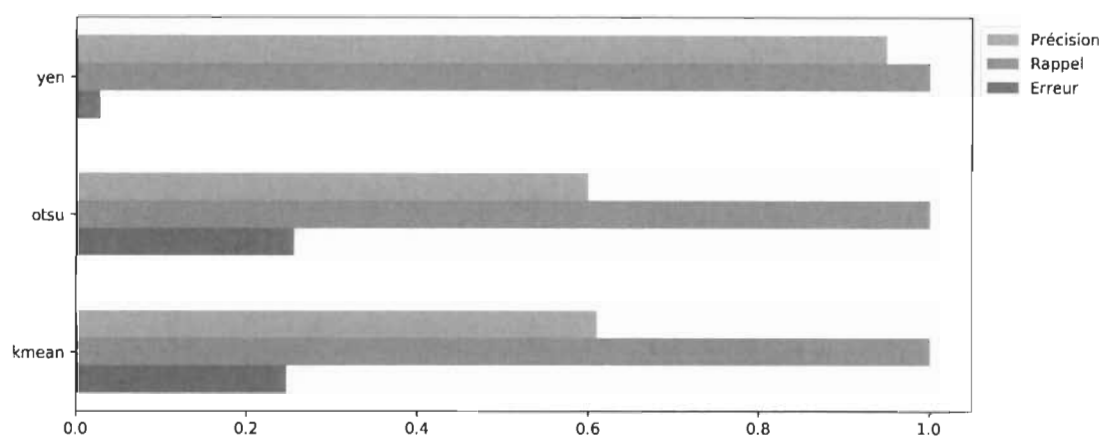


FIGURE 24 – Performance de différentes techniques de segmentation à partir de l'indice *exr*

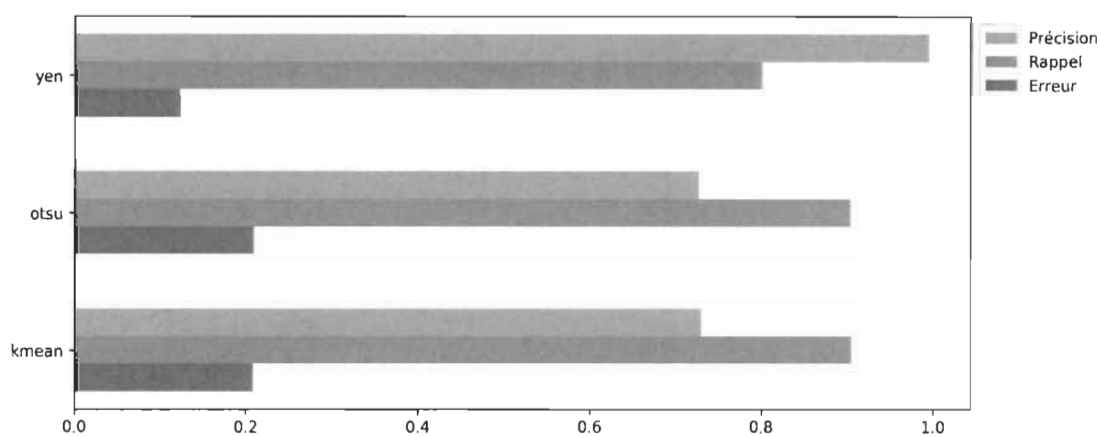


FIGURE 25 – Performance de différentes techniques de segmentation à partir de l'indice *exgr*

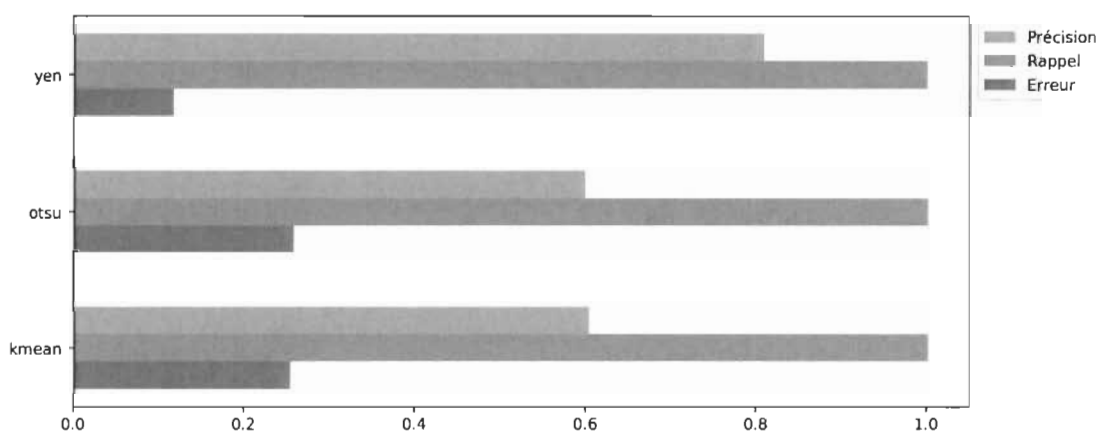


FIGURE 26 – Performance de différentes techniques de segmentation à partir de l'indice *ndvi*

Les segmentations à partir de l'indice *exg* sont celles qui ont présenté les taux d'erreurs les plus faibles. En revanche, c'est l'indice *exgr* qui détient les précisions les plus élevées. Aussi, le choix de l'indice de végétation ne semble pas favoriser une méthode de seuillage en particulier. Pour les 4 indices de végétations, la méthode Yen a été la plus précises avec un taux d'erreur relativement bas. En fait le classement des 3 méthodes de seuillage est sensiblement similaire dans les 4 graphiques.

Bien que ces 3 méthodes n'ont été testées que sur 6 extraits, un autre ensemble de mesure a été effectué. Donc au lieu de 6 parcelles choisis sur le terrain, 20 échantillons aléatoires de plus grande dimensions ont été sélectionnées. Comme pour les 6 parcelles, ces 20 échantillons ont été segmentés manuellement dans le logiciel *Gimp* pour produire une référence ou un *ground truth* pour les tests. Puis étant donné qu'il ne semble pas avoir de grande perturbation sur la performance des méthodes selon l'indice de végétation choisi, l'*exg* est pris comme base pour la segmentation.

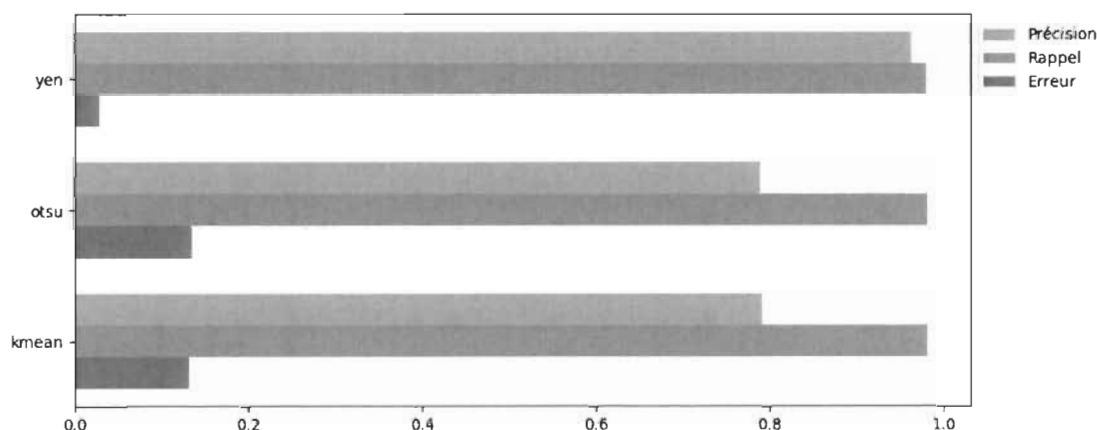


FIGURE 27 – Performance de différentes techniques de segmentation à partir de l'indice *exg* sur 20 échantillons.

Selon la figure 27, la performance des techniques de seuillage suit un patron similaire aux résultats obtenus avec les 6 parcelles. Ce qui nous confirme que la méthode Yen est définitivement la plus performante pour tous indices de végétations confondus.

4.3 Classifications avancées avec K-moyenne

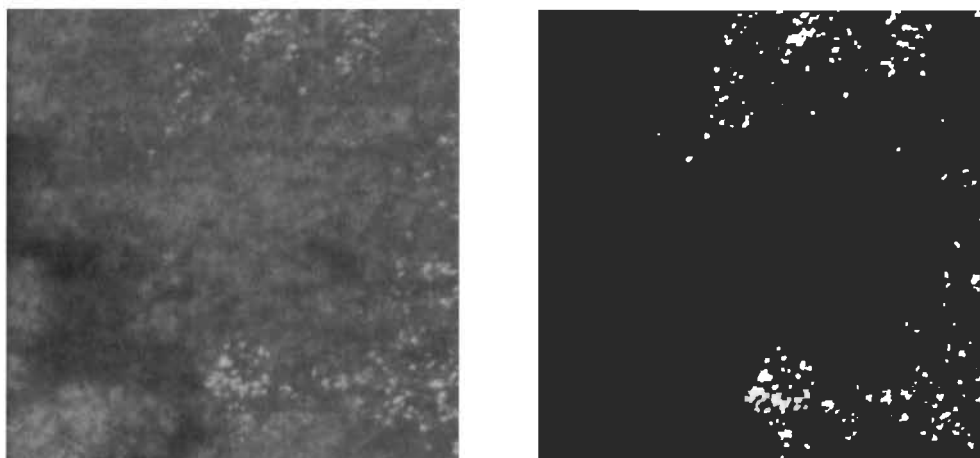


FIGURE 28 – Image avec le filtre *ndvi* et sa segmentation de référence

À la section précédente, la méthode des K-moyennes a été utilisée naïvement avec la prémisse qu'il n'y a que 2 classes d'objets dans les images. Bien que les saisons choisies favorisent le signal des mauvaises herbes avec les indices de végétations, les plants de bleuets émettent quand même un faible signal végétatif. C'est pourquoi on va varier le nombre de classes au model des K-moyennes afin de voir si on peut améliorer les performances de celle-ci. Comme à la section 4.2, les résultats vont être mesurés selon une image de référence segmentée manuellement (figure 28).

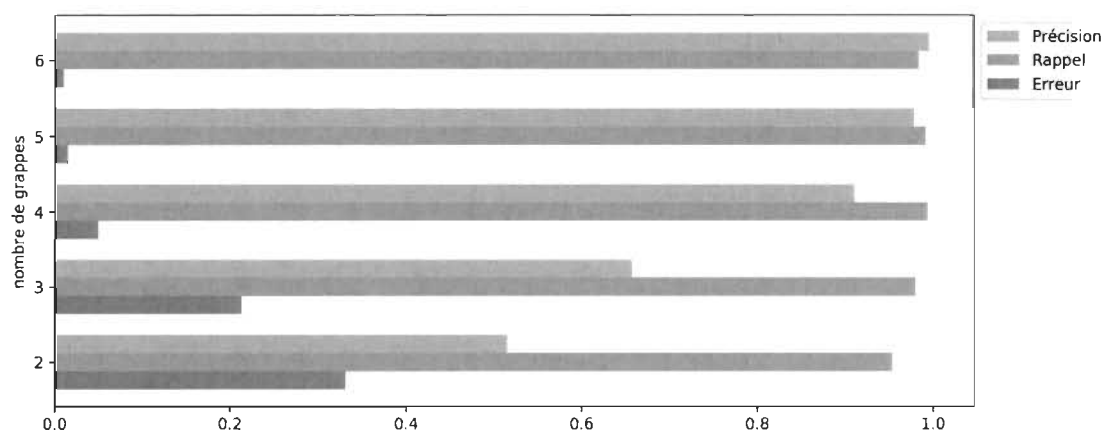


FIGURE 29 – Performance de segmentation selon le nombre de grappes

Les résultats de la figure 29 illustrent une hausse de performances significatives lorsqu'on augmente le nombre classes. Le nombre de grappes optimal à spécifier pour construire le modèle semble être 5, car au-dessus de cette valeur la performance stagne. Comme pour la section précédente, on va corroborer le constat de la figure 29 en soumettant cette nouvelle méthode aux jeux de données de 20 échantillons. Dans la figure 30, la méthode modifiée des K-moyennes a surpassée la méthode Yen.

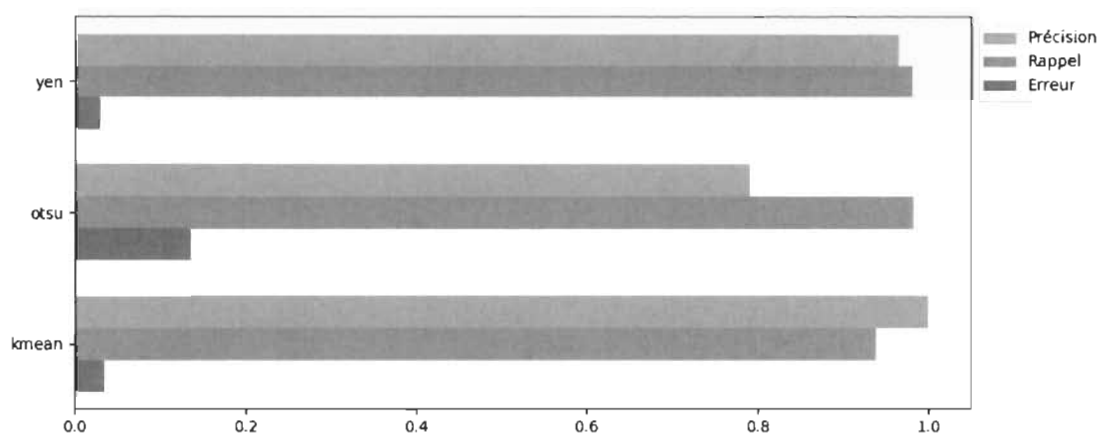


FIGURE 30 – Performance de la nouvelle méthode des K-moyennes à partir de l'indice *exg* sur 20 échantillons.

4.4 Cartographie

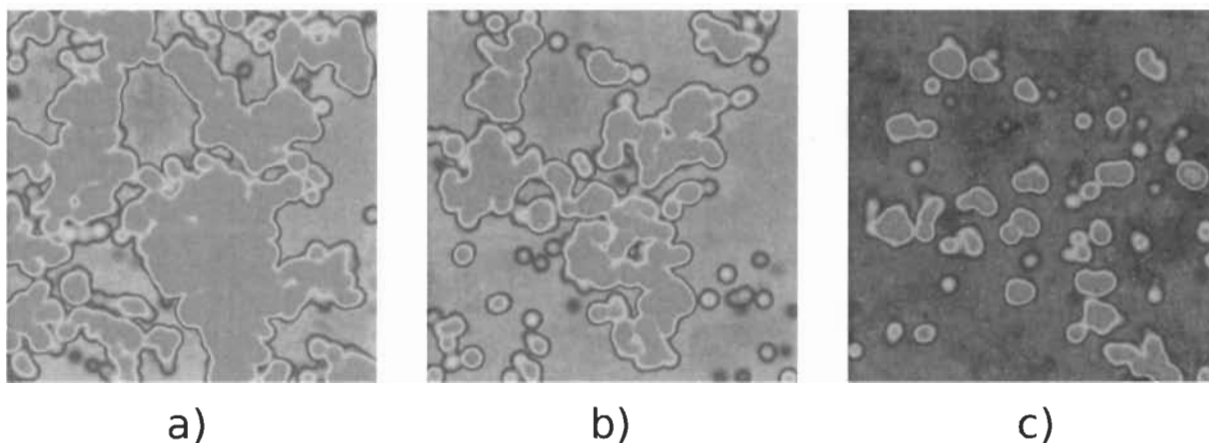


FIGURE 31 – Progression de l'infestation de la mauvaise herbe sur une période annuelle. a) 28 octobre 2019 b) 22 mai 2020 c) 9 novembre 2020

Les résultats présentés dans cette section montrent des utilisations concrètes des données obtenues dans les sections précédentes. La figure 31 montre la même parcelle sur 3 périodes différentes allant de l'automne 2019 à l'automne 2020. Avec ce type d'imagerie, il y a une facilité pour l'agriculteur de suivre la progression de la mauvaise herbe dans ses champs.

Aussi on peut aller une étape plus loin et prendre chacun des points à la figure 18 et l'image géoréférencée produite par *OpenDroneMap* pour récupérer les positions géographiques des grappes de la figure 18. L'imprime écran de la figure 32 est une carte de chaleur superposées dans une application web de cartographie.



FIGURE 32 – Carte de chaleur de la mauvaise herbe sur des images satellitaires de *Mapbox*

Les types d'utilisation ne se limite pas qu'à la production de carte de chaleur sur différents médiums. Le programme *blober* ne fait que 2 choses : trouve la position des points dans l'image et convertit ces points en positions géographiques. Il est possible de demander au programme de seulement produire les positions géographiques dans un format de fichier lisible par d'autres langages de programmation. Donc en plus de pouvoir communiquer la détection des mauvaises herbes à l'aide de visualisation, on peut aussi le communiquer à d'autres programmes.

4.5 Conclusion

Dans le chapitre des résultats, il y a eu plusieurs constats intéressants. En premier lieu, les résultats sur les techniques de segmentation selon les indices de végétations ont montré que la méthode de seuillage est plus importante que l'indice choisie. En effet, le classement de performance des techniques utilisées est demeuré identique peu importe l'indice de végétation choisie.

En deuxième lieu, les résultats des K-moyennes ont consolidé l'hypothèse émise dans le chapitre 3. L'augmentation du nombre de classes dans le modèle a permis de mieux cibler les plants de bleuets. Ainsi la soustraction des classes intermédiaires représentant les plants de bleuets est venu augmenter significativement la performance du modèle des K-moyennes. Par contre, l'ajout de dimensionnalité n'a pas aidé à la performance de celle-ci. Étant donné que l'information contenue dans les autres bandes spectrales ne reflète pas nécessairement une intensité végétative, l'ajout de cette information ne vient pas aider à la classification. Car, la valeur des centroïdes du modèle des K-moyennes perd de sa sémantique lorsqu'on ajoute une information n'étant pas directement lié au signal végétale. C'est-à-dire que les valeurs faibles de centroïdes ne correspondent plus nécessairement au sol et vice versa pour la mauvaise herbe. Alors il n'est plus possible de garantir une segmentation cohérente.

Finalement des cartes de chaleur sont présentées pour montrer une progression de l'infestation sur une période annuelle. De plus, une utilisation différente des données est illustrer avec l'utilisation d'une application web de cartographie.

Chapitre 5 Conclusion

L'objectif de ce mémoire a été d'explorer des solutions pour mieux outiller les agriculteurs du bleuet nain. Ces outils assistent l'agriculteur pour mieux le guider dans son combat contre la comptonie voyageuse et le kalmia à feuille étroite. Ce processus a passé par l'acquisition des images brutes à l'aide de drone jusqu'à l'utilisation d'application web de cartographie. L'utilisation de cette technologie va permettre aux agriculteurs de faire un contrôle plus ciblé des plants indésirables. Ainsi, les cultures conventionnelles et les cultures biologiques peuvent être à la fois plus rentables et plus écologiques. En effet, cette solution réduit l'utilisation d'herbicide et les coûts de main-d'œuvre pour le désherbage manuel.

Premièrement, ce parcours a commencé par la sélection du moment le plus approprié dans l'année pour capturer les images des champs de bleuets. Étant donné que la comptonie et le kalmia sont les premiers et les derniers plants à être relativement plus vert durant la saison végétative, le printemps et l'automne sont les moments où la capture d'image s'est réalisée. Ensuite, pour l'acquisition d'image, plusieurs logiciels ont été testés pour planifier les vols du drone et prendre en compte plusieurs paramètres afin d'optimiser la prise d'image.

Deuxièmement, il faut prendre ces collections d'images et les transformer en une orthophotographie afin d'avoir une vue d'ensemble sur le champ. Ce procédé a pu être possible avec le logiciel *OpenDroneMap* qui se spécialise dans l'analyse d'images prises par des drones.

Troisièmement, on a développé un programme pour analyser et pour transformer ces images afin d'amplifier le signal de végétation. L'intensification de ce signal s'est réalisé par l'utilisation de plusieurs indices de végétations dont l'*exg*, l'*exr*, l'*exgr* et le *ndvi*. Ensuite une variété de méthode de segmentation s'est intégrée au programme afin de pouvoir séparer la végétation du sol. L'implémentation de plusieurs de ces méthodes viennent des librairies *scikit-image* et *scikit-learn*. La méthode *Yen*, *Otsu* et *Kmean* ont été intégré afin de pouvoir comparer leurs performances.

Finalement, on utilise les métadonnées géographiques des images pour faire le pont entre les résultats de segmentation vers des applications concrètes. Par exemple, des cartes de chaleur sur 2 médiums différents ont été présentées comme application possible avec ces types de données. De plus, on laisse la porte ouverte pour d'autres applications en y exposant les données brutes de localisation dans un format digestible pour des systèmes informatiques.

Une grande part de la recherche s'est attardée sur la question de quelles méthodes est la plus efficace pour séparer le sol de la végétation. C'est pourquoi on a mesuré la performance des techniques de segmentation en variant les indices de végétation. D'une part, on a remarqué que l'indice de végétation altère légèrement les résultats des méthodes de seuillage ainsi ne favorisant aucune technique en particulier. D'autre part, la méthode *Yen* a démontré plus de précision que les autres techniques. Alors pour confirmer ce résultat, le nombre d'échantillons de test a été augmenté et les résultats ont montré un classement similaire aux autres tests. Cependant l'exploration sur la variation du nombre de classe de la méthode des K-moyennes a démontré des résultats supérieurs à la méthode *Yen*.

Il y a plusieurs cas d'utilisation pour profiter de ces informations. Notamment, les cartes de chaleurs peuvent orienter l'agriculteur dans l'arrosage d'herbicide ou dans le désherbage manuel. Ainsi il est possible de concentrer les efforts de ces activités dans les zones chaudes indiquées sur la carte. Il est aussi concevable de faire usage des données brutes de géolocalisation. Par exemple, un système de désherbage automatique sur roue pourrait utiliser les données de géolocalisation afin qu'il puisse optimiser son trajet dans les champs.

5.1 Autres approches possibles

Il existe plusieurs approches possibles pour ce type de problèmes dont l'apprentissage automatique. Une des lacunes de la solution présentée dans ce mémoire est l'isolation de beaucoup de paramètres. En effet, une grande partie de la performance des modèles présentés est dû à l'une des caractéristiques des mauvaises herbes du bleuet nain. Celles-ci restent vertes relativement longtemps durant la période végétative. C'est-à-dire qu'il est possible de l'isoler aisément avec les indices de végétation. Avec des méthodes d'apprentissage automatique

comme des réseaux neuronaux convolutifs, il serait possible de détecter une mauvaise herbe en particulier peu importe où l'on se trouve dans la saison végétative. Aussi il serait possible de faire une pierre deux coups en utilisant un modèle *RCNN* (*Region-based Convolutional Neural Network*). Cette méthode permet de classifier un objet et d'indiquer sa position dans l'image. Ainsi il serait possible d'aboutir à des résultats cartographiques similaires, mais avec de l'information additionnelle comme le type de mauvaise herbe.

D'autre part, une analyse en profondeur dans la signature des signaux des bandes multispectrales pourrait mener vers d'autres solutions. Les 3 plantes présentes dans les champs ont chacune des morphologies différentes donc nécessairement une réflexion différentes dont le proche infrarouge et le vert. Au lieu de se fonder sur les indices de végétation, un indice spécifique pourrait être développé pour une mauvaise herbe en particulier.

Références

Alessandro dos Santos Ferreira, Daniel Matte Freitas, Gercina Gonçalves da Silva, Hemerson Pistori, and Marcelo Theophilo Folhes. Weed detection in soybean crops using convnets. *Computers and Electronics in Agriculture*, 143(nil) :314–324, 2017. doi : 10.1016/j.compag.2017.10.027. URL <https://doi.org/10.1016/j.compag.2017.10.027>.

DroidPlanner. Tower. URL <https://github.com/DroidPlanner/Tower>.

dronekit. Dronekit. URL <https://dronekit.io>.

Sophie Gagnon. Essai et expérimentation sur la pollinisation et la réduction des herbicides dans la production du bleuets semi-cultivé au saguenay-lac-saint-jean. *Agrinova / Recherche et innovation en agriculture*, 2009.

Sophie Gagnon. La comptonie voyageuse. *Agrinova / Recherche et innovation en agriculture*, 2010a.

Sophie Gagnon. Le kalmia à feuilles étroites. *Agrinova / Recherche et innovation en agriculture*, 2010b.

John W Gross. A comparison of orthomosaic software for use with ultra high resolution imagery of a wetland environment. *Center for Geographic Information Science and Geography Department, Central Michigan University, Mt. Pleasant, MI, USA. Available from : http://www.imagin.org/awards/sppc/2015/papers/john_gross_paper.pdf*, 2015.

Esmael Hamuda, Martin Glavin, and Edward Jones. A survey of image processing techniques for plant extraction and segmentation in the field. *Computers and Electronics in Agriculture*, 125(nil) :184–199, 2016. doi : 10.1016/j.compag.2016.04.024. URL <https://doi.org/10.1016/j.compag.2016.04.024>.

Michael Osborne. Mission planner. URL <https://github.com/ArduPilot/MissionPlanner>.

Dong ming Li, Yuan zhi Wang, and Bo Du. Research on segmentation methods of weed and soil background under hsi color model. In *2009 Second International Workshop on*

Knowledge Discovery and Data Mining, page nil, 1 2009. doi : 10.1109/wkdd.2009.113.
URL <https://doi.org/10.1109/wkdd.2009.113>.

OpenDroneMap. Odm. URL <https://www.opendronemap.org/odm>.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, 12 :2825–2830, 2011.

Inkyu Sa, Zongyuan Ge, Feras Dayoub, Ben Upcroft, Tristan Perez, and Chris McCool. Deepfruits : a fruit detection system using deep neural networks. *Sensors*, 16(8) :1222, 2016. doi : 10.3390/s16081222. URL <https://doi.org/10.3390/s16081222>.

Jorge Torres-Sánchez, Francisca López-Granados, and José M Peña. An automatic object-based method for optimal thresholding in uav images : Application for vegetation detection in herbaceous crops. *Computers and Electronics in Agriculture*, 114 :43–52, 2015.

Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. Scikit-image : Image processing in python. *PeerJ*, 2(nil) :e453, 2014. doi : 10.7717/peerj.453. URL <https://doi.org/10.7717/peerj.453>.

Jui-Cheng Yen, Fu-Juay Chang, and Shyang Chang. A new criterion for automatic multilevel thresholding. *IEEE Transactions on Image Processing*, 4(3) :370–378, 1995. doi : 10.1109/83.366472. URL <https://doi.org/10.1109/83.366472>.